WHAT IS CLAIMED IS:

1. A software system comprising:

a server system comprising an operating system, the operating system operable to support a well-known address, the well-known address operable to receive data, the operating system further operable to provide interprocess communication;

the operating system further operable to support a buffer associated with the well-known address, the buffer comprising a finite amount of memory, and the buffer operable to store data received by the well-known address;

a plurality of handler processes operable to access a notification system in parallel, to attempt parallel acceptance of pending requests in parallel, to process error conditions, and to provide service to client requests, such that a single request received by the well-known address will result in the notification to a plurality of handler processes one of which will service the request, the remaining handler processes are operable to service other, later requests or to perform error processing if no other requests are present;

the operating system further comprising a notification system, the notification system may comprise any suitable hardware and/or software system operable to be accessed by a plurality of handler processes at any time, the notification system further operable to reflect the existence or non-existence of data in the buffer; and

a spawner process operable to create a plurality of handler processes.

2. The system of Claim 1 wherein the plurality handler processes contain a plurality of threads, wherein each thread is operable to independently handle requests.

3. The system of Claim 1 wherein the spawner process is operable to increase or decrease the number of handler processes currently in existence at any time, such operations known as load balancing.

4. The system of Claim 1 wherein the server is composed of a plurality of physical processors, each processor operable to run one or more handler processes or the spawner process.

5. A method of operating a true parallel client server system comprising the steps of:

creating a plurality of handler processes with a spawner process;

initializing a well-known address;

storing requests received by the well-known address in a buffer associated with the well-known address;

notifying, in parallel, all of the plurality of handler processes that one or more requests have arrived;

awakening any handler processes that are currently sleeping;

attempting to accept pending requests from the buffer, in parallel, with the plurality of handler processes;

servicing accepted requests with those handler processes that successfully accepted a pending request; and

processing error conditions with those handler processes that did not successfully accept a pending request.

6. The method of Claim 5 wherein attempting to accept pending requests from the buffer is also performed by a plurality of threads within the plurality of handler processes.

7. The method of Claim 5 wherein creating the plurality of handler processes with the spawner process results in the plurality of processes running on a plurality of physical processors.

8. The method of Claim 5 wherein increasing or decreasing the number of handler processes currently in existence with the spawner process, such operations known as load balancing.

9. The method of Claim 5 wherein the initialization of the well-known address is performed by cooperation between the operating system and the spawner process.

10. A software system comprising:

a server system comprising an operating system, the operating system operable to support a well-known address, the well-known address operable to receive data from a network, the operating system further operable to provide interprocess communication;

the operating system further operable to support a buffer associated with the well-known address, the buffer comprising a finite amount of memory, and the buffer operable to store information received at the well-known address;

a plurality of handler processes operable to access a notification system in parallel, to attempt parallel acceptance of pending requests from the buffer, to process error messages informing the handler process that no requests are pending in the buffer, and to service requests accepted from the buffer, such that a single request will result in the notification to a plurality of handler processes that a plurality of requests are pending in the buffer, one handler process which will service the request, the remaining handler processes operable to service other, later requests or to perform error handling if no other requests are present.

the operating system further comprising a notification system, the notification system comprising a flag, the flag operable to be accessed by a plurality of handler processes at any time, and the flag further operable to reflect the existence or non-existence of pending requests in the buffer;

a network system operable to allow communication between the server system and a plurality of client systems; and

a spawner process operable to create or destroy a plurality of handler processes, the spawner process further operable to increase or decrease the number of handler processes in existence at any moment, the magnitude of the increase or decrease based on some algorithm or heuristic, such that the algorithm or heuristic attempts to balance the number of handler processes necessary to provide rapid service to client requests with the amount of server resources used to support the handler processes.